Efficient Construction of Mesostate Networks from Molecular Dynamics Trajectories

Vitalis, Andreas ; Caflisch, Amedeo

Abstract: The coarse-graining of data from molecular simulations yields conformational space networks that may be used for predicting the system's long time scale behavior, to discover structural pathways connecting free energy basins in the system, or simply to represent accessible phase space regions of interest and their connectivities in a two-dimensional plot. In this contribution, we present a tree-based algorithm to partition conformations of biomolecules into sets of similar microstates, i.e., to coarse-grain trajectory data into mesostates. On account of utilizing an architecture similar to that of established tree-based algorithms, the proposed scheme operates in near-linear time with data set size. We derive expressions needed for the fast evaluation of mesostate properties and distances when employing typical choices for measures of similarity between microstates. Using both a pedagogically useful and a real-word application, the algorithm is shown to be robust with respect to tree height, which in addition to mesostate threshold size is the main adjustable parameter. It is demonstrated that the derived mesostate networks can preserve information regarding the free energy basins and barriers by which the system is characterized.

DOI: https://doi.org/10.1021/ct200801b

Efficient Construction of Mesostate Networks from Molecular Dynamics Trajectories

Andreas Vitalis,[1,*] and Amedeo Caflisch[1,*]

[1]*Department of Biochemistry*

University of Zurich

Winterthurerstrasse 190, CH-8057 Zurich, Switzerland

*: To whom correspondence should be addressed:

Amedeo Caflisch: Tel: +41446355521, E-mail: caflisch@bioc.uzh.ch

Andreas Vitalis: Tel: +41446355597, E-mail: a.vitalis@bioc.uzh.ch

ABSTRACT

The coarse-graining of data from molecular simulations yields conformational space networks that may be used for predicting the system's long timescale behavior, to discover structural pathways connecting free energy basins in the system, or simply to represent accessible phase space regions of interest and their connectivities in a two-dimensional plot. In this contribution, we present a tree-based algorithm to partition conformations of biomolecules into sets of similar microstates, *i.e.*, to coarse-grain trajectory data into mesostates. On account of utilizing an architecture similar to that of established tree-based algorithms, the proposed scheme operates in near-linear time with dataset size. We derive expressions needed for the fast evaluation of mesostate properties and distances when employing typical choices for measures of similarity between microstates. Using both a pedagogically useful and a real-word application, the algorithm is shown to be robust with respect to tree height, which in addition to mesostate threshold size is the main adjustable parameter. It is demonstrated that the derived mesostate networks can preserve information regarding the free energy basins and barriers the system is characterized by.

INTRODUCTION

Clustering or coarse-graining of molecular simulation data through measures of geometrical or kinetic similarity is a special case of a broad class of problems in data analysis.[1] Clustering of molecular trajectory information is used most often to identify free energy basins and the structural pathways connecting them,[2–4] but can also serve to estimate entropy (occupied phase space volume),[5] check for simulation convergence,[6–9] or simply to condense trajectory information to highlight qualitative trends and features of an ensemble.[10,11] The use of geometric clustering to identify fine-grained mesostates[a] constituting a conformational space network at equilibrium is a very powerful technique as these networks in principle are able to represent both thermodynamics and kinetics of the system in detail.[12–16]

Computing these networks requires that simulation data are sampled frequently enough to resolve transitions of interest between those mesostates that exchange most rapidly. This is linked to the chosen resolution of mesostates. Otherwise, shortcuts are introduced and processes will no longer be resolved or be described inaccurately at the kinetic level.[17] Due to the sheer number of terms, convergence of transition probabilities often requires frequent recording of trajectory information, which potentially gives rise to very large datasets.[18,19] When using geometric criteria for clustering,[20] mesostates should not differ drastically in phase space volume (resolution) since conformational diffusion in the absence of significant barriers sets a fundamental timescale. If for example one were to combine all "unstructured" states into a single, "entropic" mesostate, kinetics of pathways passing through such a significantly larger mesostate will be incorrectly described at the network level. This is because the real physical pathway, which involves different subpopulations of the mesostate that are not reachable within the fundamental time step, is now masked. Similarly, mesostate centers should be placed preferably at regions of high density (basins) in order to minimize the risk of crossing low-density regions (barriers), which may be geometrically narrow, within a single mesostate.

From these explicit or implicit requirements, we can derive the following demands toward a clustering algorithm for molecular simulation data. We look for an algorithm that operates in linear time, handles large data sets of high dimensionality, does not impose any specific *a priori* partitioning criteria (whether in the number of mesostates or the boundaries connecting them), yields homogeneous cluster volumes, chooses cluster centers in accordance with local density, and keeps cluster overlap minimal. Specifically for the identification of mesostate networks, we do not require that all points within a

a We will refer to individual trajectory snapshots as microstates, and to collections of similar microstates identified by a clustering algorithm as mesostates.

region of homogeneous density belong to the same cluster, or that the clustering is exactly stable, *i.e.*, weak input order dependence is tolerable if the number of mesostates is large. Furthermore, none of the input data is interpreted as database "noise", and algorithms relying explicitly on database sampling (those that try to derive mesostates by considering only a subset of the data) are not of interest. These last three points may be altered if the goals are different, *e.g.*, they lie in identifying few geometric clusters.[21]

It should be emphasized that the algorithm derived in this work is a general data processing tool, and need not be restricted to the application domain chosen in this work. For the latter, dedicated simulation and analysis schemes have been developed[22,23] and applied successfully.[24,25] These explicit path sampling schemes rely on nonequilibrium sampling of transitions between local minima identified independently, *e.g.*, as inherent structures.[26] They may overall be more efficient, and in a second step they often allow straightforward grouping (lumping) of states (minima) according to a threshold timescale.[27] Advanced sampling methodologies may yield improved overall efficiency because sufficient sampling of low likelihood regions of phase space will often be difficult to attain using conventional molecular dynamics. The literature offers similar, kinetic (re)grouping techniques that operate on fine-grained mesostate networks obtained from structural clustering.[16,19,28]

In clustering, the issue of dimensionality deserves particular attention since very frequently molecular simulation data are represented in fairly high-dimensional spaces ($D \approx 100 - 1000$). The so-called "curse of dimensionality"[29] is a colloquialism for the fact that high-dimensional spaces generally lead to low data density (sparsity) due to exponential growth of the available space. This effect is most pronounced if all dimensions are decoupled sources generating white noise. For molecular systems in Cartesian space, however, the covalent topology alone will exclude the vast majority of said space on account of manifold correlations between the chosen degrees of freedom. This is the reason why – for example - it often makes little difference to use C_α atoms only *vs*. all backbone heavy atoms for clusterings using the positional root mean square deviation (RMSD) of aligned coordinates despite dimensionality increasing by a factor of 4. Mismatches in apparent and actual dimensionalities can sometimes be addressed by the use of degrees of freedom that do not experience strong topology-derived correlations, *e.g.*, dihedral angles. However, data sparsity continues to become critically low if too many weakly coupled dimensions are part of the chosen coordinate space, for instance when including both intramolecular- and intermolecular degrees of freedom, or when including sidechain conformations. Then, measures of distance that respect the full dimensionality will show a spectrum that is almost

entirely depleted for small values, and exhibits an increasingly narrow distribution otherwise.[30] This contraction of minimum and maximum observed distances is a well-known phenomenon, and essentially renders neighbor relations arbitrary.[31] In simulation terms, this is a manifestation of sampling problems related to the combinatorial complexity of weakly coupled processes. For instance, if there are 15 independently moving sidechains with three rotamers each, there are already in excess of 1.4×10^7 possible configurations, *i.e.*, a number exceeding the size of typical datasets. The key here is that one will typically consider those processes to be sufficiently independent of one another, such that recurrent sampling of each sidechain is deemed to be enough. For clustering this means that it is not permissible to blindly include all of them in the coordinate subset, because the conformational distances caused by the set of weakly coupled processes will drown out signals coming from processes of interest. Instead, these motions are usually discarded entirely, and this is called feature selection in the data processing literature.[32]

Aside from relying on massive dimensionality reduction using principal components[33] or other techniques,[18,34,35] current state-of-the-art in the field is to use algorithms for mesostate identification that present a reasonable compromise between efficiency and robustness. It is undesirable to have many parameters or system-specific performance characteristics because the structure of the data is not known *a priori*. The simplest class of approaches is based on the Leader algorithm[36] (in Prinz *et al.*[37] referred to a regular space clustering).[38,39] Alternatively, fixed partition algorithms such as the approximate *K*-centers (*K*-medoids) algorithms[40,41] are in use.[15,19,37,42] All aforementioned methods scale superlinearly with dataset size (because in fixed partitioning schemes, the number of mesostates, *K*, will have to be proportional to *N* unless sampling is exhaustive). Post-processing of initial results may involve application of similar or more rigorous algorithms such as strictly hierarchical schemes.[43–45] The reason for using simple algorithms appears to be solely that they are reasonably affordable in both memory and CPU time for large datasets. It has been argued that at the level of coarse Markov state models, details of the algorithm are not important.[37] Note again that the aforementioned algorithms are general data processing tools with a modular definition of similarity, and that dedicated grouping schemes as discussed may be available.[19,27,28]

In this contribution, we propose a tree-based algorithm that relies on partitioning according to a preset schedule of threshold criteria operating at each level of the tree. Clusters or mesostates at coarser resolution serve as parent nodes to a set of mesostates at the next finest level. Inherently a multi-resolution technique, the algorithm utilizes the parent-child relations to limit the search space for the

branches, and thereby achieves near linear scaling with dataset size. The height of the tree is a fixed parameter that can be tuned to optimize computational cost. The algorithm is architecturally similar to the BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) and related clustering algorithms.[46,47] Given that the motivation behind those tree-based algorithms is fundamentally different (optimization of spatial demands to hardware limitations), the proposed scheme deviates substantially in most of the actual implementation. The rest of this article is structured as follows. After describing the algorithm itself in detail, we address the modifications necessary to evaluations of conformational distances when typical descriptors of molecules such as Cartesian coordinates or dihedral angles are used. The accuracy of these approximations is discussed, followed by an illustration of the performance of the proposed algorithm on a 2-dimensional dataset. Using datasets of realistic dimensionality, the impact of tree height on clustering results and efficiency are explored, and a scaling analysis with dataset size is performed. Lastly, we use network-derived properties to analyze the robustness of the extracted mesostate networks for a simple and pedagogically useful system as well as a realistic test case. Based on our results, we conclude that the algorithm represents a good compromise between efficiency and quality of the derived mesostate networks.

METHODS

Description of the Algorithm

Consider a pseudotree of height H with an associated vector of threshold values $t_1,t_2,..t_H$ (see Fig. 1). The top (root) will (formally) consist of exactly one parent node containing the entire dataset. We will process the data in arbitrary order (sequentially for simplicity), and for each point j scan a cluster set $\{c_k\}$ at each level k from H down to 2 (1 being the leaf level). If $\{c_k\}$ contains a cluster c_k^m such that $d_{CP}(c_k^m,j)$ is less than t_k, we store the respective cluster index m_k, and add snapshot j to $c_k^{m_k}$. This will change the centroid of $c_k^{m_k}$, which in turn affects all distance evaluations (also retroactively). Centroids drift toward regions of high data density, which implies that the threshold criteria are rarely fulfilled exactly for all snapshots that have been added to a cluster, in particular at the higher levels. If k equals H, $\{c_k\}$ is the set of all clusters at that level (since there is only a single root), otherwise it is the set of children of $c_{k+1}^{m_{k+1}}$. If at any given level the search is unsuccessful, the corresponding k is recorded, and the cluster list at the next lower level will instead consist of the children of the c_k^m that had the smallest $d_{CP}(c_k^m,j)$. Any assignment failure will lead to new clusters being spawned for all undefined m_k's. For consecutive, undefined m_k's (the most common case), the resultant clusters will of course all consist of

the same snapshot and have identical centroids. Occasionally, a failed assignment will recover at a level higher than 2. This is because children of a cluster can extend beyond the threshold radius set for the parent cluster. Then, the identified c_k^m will be made a child also of the newly created c_{k+1}^l, *i.e.*, it effectively has two parents. This violates the definition of a tree (hence pseudotree), but is i) usually rare, and ii) irrelevant algorithmically since we never attempt to follow a path from the leaves to the root.

After the entire dataset has been scanned once, we have a fixed tree with a set of clusters $\{c_2\}$. The second scan of the input data works identically for levels H to 2 with the exception that the snapshots are not actually added to the clusters (and therefore the centroids at these levels are static). Note, however, that it is still possible for snapshots to remain unassigned at one of those levels (due to combination of a specific input order and centroid drift). Most importantly, in the second pass we also descend to the leaf level (1) by searching the children of $c_2^{m_2}$ (defined identically) that are now being created. If no children exist yet, or if no cluster is found for which $d_{CP}(c_1^m, j)$ is less than t_1, a new cluster at level 1 is spawned, otherwise an existing leaf cluster is appended by the current snapshot. An example tree illustrating the verbiage above is provided in Fig. 1.

The efficiency of the algorithm relies on the fact that for a range of settings the number of clusters scanned per snapshot is approximately constant. For higher trees, the average number of children per level decreases, which compensates the cost incurred by having to consider more levels. For $H = 1$, the algorithm relaxes to a simple Leader-like algorithm with centroid drift. Efficiency is also impaired if t_H approaches t_1.

Distance Computations

To keep the computations feasible, we utilize the clustering feature (CF-)vector introduced in the BIRCH algorithm.[46] CF-vectors are incremented by each added snapshot, and contain mean information regarding the mesostate, specifically the linear sum $\vec{\mathrm{LS}}_c$, squared sum, SS_c, and number of snapshots, N_c. For Euclidean (L_2) distance measures, CF-vectors allow rapid calculation of a variety of cluster properties and inter-cluster distances:

$$\begin{aligned} &\vec{\mathrm{LS}}_c = \sum_i^{N_c} \vec{\mathrm{S}}_{k(i)} \quad \text{and} \quad SS_c = \sum_i^{N_c} \vec{\mathrm{S}}_{k(i)}^{\,2} \\ &r_c^2 = \left(1/N_c^2\right)\left(N_c SS_c - \vec{\mathrm{LS}}_c^2\right) \\ &d_c^2 = \left(N_c(N_c-1)/2\right)^{-1}\left(N_c SS_c - \vec{\mathrm{LS}}_c^2\right) \end{aligned} \qquad (1)$$

Here, $\vec{\mathrm{S}}_j$ is the data vector of snapshot j, and r_c, and d_c are cluster radii and diameters, respectively.

r_c corresponds to the mean distance of snapshots from the centroid, and d_c to the mean distance between snapshots. The centroid ($\vec{\mathrm{LS}}_c$ /N_c) is readily available as well. For the critical computation of $d_{CP}(c_k^m,j)$, we can choose different models including the simple centroid-centroid distance (d_{CC}) or the mean pairwise distance between members of different mesostates (d_{IC}). Both measures generalize to the case where one of the two clusters is only a single snapshot, *i.e.*, the case required by $d_{CP}(c_k^m,j)$, and the formulas are as follows:

$$\begin{aligned} d_{CC}^2(c_A, j) &= \left(N_A^{-1}\vec{\mathrm{LS}}_A - \vec{\mathrm{S}}_j\right)^2 \\ d_{IC}^2(c_A, j) &= N_A^{-1}\left(SS_A + N_A\vec{\mathrm{S}}_j^2 - 2\,\vec{\mathrm{LS}}_A\cdot\vec{\mathrm{S}}_j\right) \end{aligned} \qquad (2)$$

d_{IC} and d_{CC} become increasingly similar when the distance between centroid and snapshot gets larger relative to the cluster radius. Conversely, for distances on par with cluster radius, d_{IC} will generally be larger than d_{CC}. The two values are identical if N_A is 1. We generally choose to normalize all distances by the dimensionality, D (or by D/3 in the case of the Cartesian coordinates), which means that the formulas for the squared quantities in equations 1 and 2 need to be extended on the right-hand side by a corresponding factor, typically D^{-1}.

We utilize the CF-vector to be able to quickly evaluate relative and internal cluster properties. Unlike in the BIRCH algorithm, we do not attempt to condense the dataset into CF-vectors to satisfy a spatial constraint. Instead, we do maintain a list of snapshots added to each mesostate to be able to later derive the corresponding transition network.

Adaptation to Typical Data from Molecular Simulations

Root-mean-square deviation of atomic positions (RMSD): When considering the RMSD of atomic positions, $\vec{\mathrm{X}}$, between snapshots as the fundamental measure of distance, it is usually implied that the two sets of data are aligned prior to RMSD computation:

$$d^2(i, j) = 3\,D^{-1}\left[\vec{\mathrm{X}}_i - \mathrm{O}_T\left(\mathrm{O}_R\left(\vec{\mathrm{X}}_j\right)\right)\right]^2 \text{ with } \mathrm{O}_T \text{ and } \mathrm{O}_R \text{ chosen such that } d^2(i, j) \text{ min.} \qquad (3)$$

In equation 3, the $3D^{-1}$ term is the aforementioned normalization by dimensionality as is implied in the definition of RMSD. The translational operator, O_T, is obtained by overlapping the centroids, and the rotational operator, O_R, can be determined exactly by a quaternion method. Operators will be unique for each pair of snapshots implying that the definition of the CF-vector becomes nontrivial. We use the following heuristic to solve this issue. Computing the values for $d_{CP}(c_k^m,j)$ utilizes alignment of snapshot j to the current centroid of cluster c_k^m. When adding snapshot j to cluster c_k^m, appending

$\vec{LS}_c$ and SS_c is preceded by an identical alignment. Fixed weights can be added to this computation (*e.g.*, atomic masses) as long as they are also used for alignment, and as long as data are centered first. The proposed heuristic is expected to fail whenever considerably heterogeneous sets of snapshots are involved, *e.g.*, when evaluating d_{IC} for two large and well-separated clusters.

Periodic data such as dihedral angles: For clustering directly in dihedral angle space,[48] the periodicity of the underlying data becomes problematic. We can uniquely define a distance between two vectors of dihedral angles, $\vec{\Phi}_i$ and $\vec{\Phi}_j$, corresponding to two microstates:

$$d^2(i,j) = D^{-1}\left[\left(\vec{\Phi}_i - \vec{\Phi}_j\right) \bmod 2\pi\right]^2 \tag{4}$$

Essentially, each distance evaluation requires a check as to which periodic image (frame) is nearest. Correspondingly, definitions of centroids are altered, and variances are no longer uniquely defined. To be able to continue to use the simple equations above, we therefore have to modify the way the CF-vectors are incremented.

$$\begin{aligned}
&\vec{LS}_{new} = \vec{LS}_t - N_{new}\,\vec{A} \\
&SS_{new} = SS_t + N_{new}\,\vec{A}^2 - 2\,\vec{A}\cdot\vec{LS}_t \\
&\vec{LS}_t = \vec{LS}_{old} + \vec{\Phi}_j - 2\pi\left[W(\vec{\Delta}_\phi) - 1\right] \\
&SS_t = SS_{old} + \left(\vec{\Phi}_j - 2\pi\left[W(\vec{\Delta}_\phi) - 1\right]\right)^2 \text{ with} \\
&W(x) = H(x-\pi) + H(x+\pi) \text{ and } \vec{\Delta}_\phi = \vec{\Phi}_j - \vec{LS}_{old}/N_{old} \text{ and } \vec{A} = 2\pi\left[W(\vec{LS}_{new}\,N_{new}^{-1}) - 1\right]
\end{aligned} \tag{5}$$

Here, we assume that a snapshot with index j is added to an existing cluster of size N_{old}. H(x) denotes the Heaviside function, and N_{new}=N_{old}+1. The procedure is essentially a two-step process that first shifts the snapshot into the right periodic image, and after addition corrects for possible boundary violations in the linear sum itself (note that dihedral angles are assumed to be defined on the interval [-π,π]). It is important to point out that the added overhead for each added snapshot is of $O(D)$ only. The treatment is approximate for cluster diameter (see Table 1) because the shift vectors $\vec{A}$ are defined as pseudo-averages at the centroid level, where in reality they are truly pairwise terms.

Data with fluctuating weights: Fluctuating weights mean that the contribution an individual degree of freedom makes to the evaluation of distance between data instances can change throughout the dataset. In this contribution, we test whether the simple algebraic transformations utilizing CF-vectors can be extended to such a case. As an example, we propose a set of dihedral angles for clustering, for which the weight corresponds to the sum of conformation-dependent moments of inertia associated with the dihedral angle in the two respective conformations:

$$d^2(i,j)=b\left(\left[\vec{\mathrm{I}}_i+\vec{\mathrm{I}}_j\right]\cdot\left[\vec{\mathrm{V}}_c(i,j)+\vec{\mathrm{V}}_s(i,j)\right]\right) \text{ with } b=\left(2\sum_k^D\left[I_i^k+I_j^k\right]\right)^{-1}$$
$$\vec{\mathrm{V}}_c(i,j)=\left(\cos\vec{\Phi}_i-\cos\vec{\Phi}_j\right)\circ\left(\cos\vec{\Phi}_i-\cos\vec{\Phi}_j\right) \quad (6)$$
$$\vec{\mathrm{V}}_s(i,j)=\left(\sin\vec{\Phi}_i-\sin\vec{\Phi}_j\right)\circ\left(\sin\vec{\Phi}_i-\sin\vec{\Phi}_j\right)$$

Here, I_i^k is the k^{th} element of moment of inertia vector $\vec{\mathrm{I}}_i$, and "$\circ$" denotes the element-by-element product. Equation 6 chooses the sine/cosine terms of the angles to eliminate explicit periodicity. The vectors $\vec{\mathrm{I}}_i$ need to be stored along with the dihedral angles. Note that dimensionality normalization is handled through the first term on the right-hand side of equation 6. An extended CF-vector is required to derive approximate relations for cluster radii, etc. These are provided in the Supporting Methods. Equation 6 is just one possible choice for a set of fluctuation weights, and ongoing research concerns the identification of alternative weights for the basic coordinates microstates are represented in.

Refinement

The clustering that is obtained initially may be refined. For instance, we can consider merging clusters that are adjacent using an appropriate heuristic. We could also attempt to remove cluster outliers to tighten clusters, or to re-cluster snapshots that form a "cluster" just by themselves. The important point is that none of those procedures are unique to the algorithm or required by our goals, and that they typically exhibit unfavorable scaling with dataset size. The only explicit refinement considered during development was to merge clusters that would yield either a reduced joint diameter or radius relative to the respective weighted averages of the original clusters. Numerical tests showed that these criteria are rarely fulfilled for the unrefined results. Therefore, refinement is not considered further in this article.

Datasets

DS1: Data are derived from recently published simulations on the intrinsically disordered peptide $A\beta_{12\text{-}28}$.[39] The trajectory contained $2.5\text{x}10^4$ snapshots saved at an interval of 20 ps. 144 internal distances between backbone nitrogen and oxygen atoms of sufficiently spaced, nonterminal residues served as the input data for extracting principal components via the discrete Karhunen-Loève transform. After transforming the entire dataset, the two components with the largest variances were isolated and served as *DS1*.

DS2: Data are derived from recently published simulations on the intrinsically disordered peptide $A\beta_{12\text{-}28}$.[39] The trajectory contained $7.5\text{x}10^5$ snapshots saved at an interval of 20 ps, and the same 144, partially

redundant internal distances that DS1 was originally derived from were extracted at each frame ($D = 144$). For the data in Table 1, backbone ($\phi/\psi/\omega$) dihedral angles of residues 14-24 of the peptide ($D = 66$), their sine/cosine terms ($D = 66$), the respective terms weighted by inertial masses (see equation 6), or the Cartesian coordinates of backbone nitrogen and oxygen atoms of residues 14-24 ($D = 66$) were also extracted.

DS3: The source of the data is the same as for *DS2* only with a combined trajectory with data from multiple simulations (up to 6×10^6 snapshots). The coordinate subset extracted from each frame were the sine/cosine values of the backbone ($\phi/\psi/\omega$) dihedral angles of residues 14-24 of the peptide ($D = 66$).

DS4: The small molecule *n*-butane was simulated at all-atom resolution in the presence of completely constrained bond lengths and angles, and with dihedral angle potentials derived from the OPLS-AA force field[49] as the only term in the Hamiltonian. This effectively uncouples the three torsional degrees of freedom. At 400 K, a continuous trajectory of 50000 snapshots was obtained spaced at 36 fs. Since all atoms are assumed to be labeled, degeneracies due to identical hydrogens are removed. Each dihedral angle potential has threefold symmetry (*anti* or *a*, *gauche*$^+$ or g^+, *gauche*$^-$ or g^-) allowing the identification of $3 \cdot 3 \cdot 3 = 27$ coarse states.

DS5: The miniprotein beta3S folds into a three-stranded β-sheet. At 330K, reversible folding is observed reliably as shown in prior work.[28] *DS5* was generated from a total of 20 μs of simulation time containing 1×10^6 snapshots saved at an interval of 20 ps.

RESULTS

Below, we present two categories of results. The first three subsections are concerned with the algorithm itself, *i.e.*, its qualitative performance, its efficiency and parameter sensitivity, and the accuracy of the derived approximation formulas. The remaining three subsections examine the utility of the proposed scheme specifically in the context of constructing fine-grained mesostate networks, and whether those networks appear to preserve information regarding free energy basins and barriers. The latter makes use of both a simple, but pedagogically useful example and of a real-world application.

Accuracy of simplified computations based on CF-vectors

Table 1 shows how accurate it is to use the simplified formulas derived above and in the Supporting Methods. For D-RMS and sine/cosine transforms of dihedral angles, the formulas are exact to machine precision. In all other cases, accuracy suffers. For dihedral angles, cluster radii are exact, while

diameters and in particular d_{IC}-values suffer. The error gets large if the periodic shift vectors become increasingly heterogeneous and are ill-approximated by vector $\vec{A}$ in equation 5, *i.e.*, results deteriorate with increasing distance values and increasing dimensionality. Conversely, the approximations made to be able to treat fluctuating weights have a rather uniform impact – at least for the case studied here. Lastly, RMSD values of Cartesian coordinates are somewhere in the middle. Here, larger underlying distances lead to more heterogeneity in the alignment operators, which in turn leads to maximally decreased accuracy for d_{IC}. It is important to point out that it is possible to run the algorithm described here without ever considering d_c, r_c, or d_{IC} by choosing d_{CC} to decide whether to assign a snapshot to an existing cluster. The derivations are needed primarily to permit computation of cluster properties and refinement operations with time complexities that do not exceed that of the algorithm itself.

Qualitative evaluation of proposed algorithm in comparison to reference methods

Fig. 2 shows how the proposed algorithm works in comparison to two other clustering algorithms. The dataset considered (*DS1*) is shown as a scatter plot in Panel A along with contour lines. There are two dominant basins embedded into relatively uniform low-density regions. These data are a realistic representation of analysis of molecular simulation data in low-dimensional projections. Qualitatively, drawing boundaries to delineate basins is challenging given the structure of the data. Panels B-D show the 30 largest clusters from the proposed algorithm with $H=4$ (B), from the simple Leader algorithm (C), and from a rigorous, agglomerative clustering (D) using a mean linkage criterion (see Supporting Methods). All algorithms identify mesostates in accordance with regions of high density. Mesostates appear largest for the Leader algorithm (C), and smallest for the proposed scheme (B). Cluster shapes are distinctively noncircular, in particular for Panels B and D. For the Leader algorithm, the suboptimal assignment of mesostate centers (see Supporting Methods) gives rise to overlap and mesostates with small occupied volumes. Mesostate boundaries are of arbitrary shape in the agglomerative scheme (D), curved in the Leader scheme (C), and more or less linear for the proposed algorithm (B). In summary, Fig. 2 shows that the results from the proposed algorithm with $H=4$ provide a qualitatively similar picture to those from a rigorous, agglomerative algorithm. Fig. S1 highlights the origin of the linear mesostate boundaries in the former, and shows that large values of H can give rise to undesirable effects for these low-dimensional data.

Parameter-dependence and scaling properties

Fig. 3 shows that the increase in the number of mesostates reported in Figs. 2 and S1 is a systematic

function of the chosen tree height H. For the given example (*DS2*), the total number of mesostates is constant when considering only those constituted by at least 10 microstates. This means that the increase is primarily a result of failing to cluster microstates in low-density regions with the strongest contribution coming from resultant "meso"states of size 1 (Fig. 3). In essence, an increasing number of levels will – in data-dependent fashion – create more and more dividing lines between regions of data space (see Fig. S1). For two microstates that are within a normalized distance of t_1 of one another, those dividing lines will eventually lead to a divergence in the paths taken through the tree. It remains to be seen whether this H-dependency poses a problem beyond having to slightly renormalize the chosen value for t_1. The renormalization is manifested in Panel B of Fig. 3 that shows a more or less linear decrease in the average snapshot-centroid distance with increasing H. In terms of computational complexity, the algorithm clearly has a minimum as a function of H (Panel B of Fig. 3). An initial and strong decrease in computational cost crosses over into a regime where CPU time increases linearly with H. We typically employ values of H ranging from 4-24 depending on the dataset. Naturally, we also examined the scaling of the algorithm with dataset size, and these results are shown in Fig. 4. A linear dependence on dataset size is observed as expected, and the proposed scheme outperforms the (superlinear) Leader algorithm substantially for large dataset sizes. Regarding the dependency on H, for the data in Fig. 4 it sufficed to use a fixed value of 16 throughout. This dispels concerns regarding parameter-dependent efficiency in the application of the proposed algorithm to real molecular simulation data in high-dimensional spaces.

Quantitative comparison of proposed algorithm in comparison to reference methods

Next, we wish to analyze whether the algorithm introduces artificial features to the derived mesostate network. It is unfortunately difficult to convert such a network into a quantitative and informative readout. Tests of Markovianity[14,37,50,51] or diffusivity[52] report on whether the network satisfies specific properties, but failure statistics of those tests are poor quantitative descriptors of the networks themselves. Here, we employ cut-based free energy profiles (cFEPs) utilizing the mean-first passage times ($mfpt$)[17,53] to a chosen reference mesostate (c_{Ref}) to partition the network into two components. The number of transitions between these two components is the partition function of the cut (Z_{AB}), and can be semi-quantitatively related to a free energy. For each mesostate i, its $mfpt_i$ can be used to define the cut between two partitions with either smaller or longer $mfpt$-values, and the cumulative probability density of all mesostates with $mfpt_j<mfpt_i$ can be used as the associated progress variable. An alternative would have involved using free energy disconnectivity graphs[54,55] that depict the structure of

the free energy landscape as a hierarchical graph. However, cFEPs allow clearer quantitative comparisons between multiple networks.

The toy system we use to evaluate the algorithm is labeled *n*-butane meaning that all hydrogens are distinguishable. Exact constraints on bond lengths and angles mean that the effective $D = 3$ regardless of the chosen microstate representation. Each of the three dihedral angles has three basins (*a*, g^+, g^-) with the one around the central C-C bond favoring the *anti*-conformation over the two *gauche* states, while the two C-C-C-H torsions populate all three states with equal likelihood. Kinetic distances are expected to show large overlap, *e.g.*, with state ag^+a being equally far away from state *aaa* as states ag^-a, aag^+, or aag^- (the first character denoting the rotation around the central C-C bond, the latter two that around the two terminal C-C bonds). Fig. 5 demonstrates that the cFEPs for this system are independent both of the algorithm used to obtain mesostates and of the chosen representation (dihedral angles in Panel A or RMSD in Panel B). Differences between algorithms are hardly significant, because they stem from reordering of minor basins that overlap kinetically and from differences in the amount of overlap resolved. The former is seen generally for the basins where C-C-C-C is not *anti*, while the latter can be observed for instance in Panel A for the proposed scheme with $H = 24$, where at $Z_A/Z \approx 0.52$ the barrier separating ag^+g^+ from $aag^{+/-}$ is eliminated. Importantly, the three main barriers, *i.e.*, the one separating the first basin from all the rest ($Z_A/Z \approx 0.11$), the one separating states accessible by one methyl rotation from those accessible by two methyl rotations ($Z_A/Z \approx 0.39$), and the one separating *gauche* from *anti* states for the central torsion angle ($Z_A/Z \approx 0.66$), are all quantitatively invariant for all cFEPs shown in Fig. 5.

This congruence is seen despite the fact that mesostate volumes and numbers differ significantly between algorithms (see Table 2). Consistent with Fig. 2, the data in Table 2 show that the phase space partitioning obtained for the proposed algorithm does not suffer from mesostate overlap irrespective of the chosen *H*. If overlap were a significant factor, one would expect the apparent phase space volume coverage to be correspondingly larger than for the rigorous agglomerative scheme. Instead, Table 2 makes the point that the proposed algorithm is roughly on par with the agglomerative scheme, and clearly outperforms the Leader algorithm in this regard. The differences in the numbers of mesostates do have quantitative impact, *viz.*, in the actual *mfpt*-values. However, Fig. S2 shows that at least in this particular case the changes are very systematic, and correspond to an overall shift in the *mfpt*-distributions.

Improved performance on a real-world example

Lastly, we examine a realistic test case (*DS5*). At 330 K, the miniprotein beta3S folds reversibly into a three-stranded β-sheet topology on the high ns-timescale when using a particular computational model as discussed in prior work.[28] Unfortunately, discussing this system's free energy landscape and its intricacies[17] would go substantially beyond the scope of this article. Fig. 6 shows cFEPs from a mesostate that is part of the folded basin. The cFEPs are structurally annotated by secondary assignments according to DSSP distinguishing 5 variants of β-secondary structure, 3 types of helices, turn-like conformations, and highly curved (bent) regions. Panel A of Fig. 6 shows data based on RMSD-based clustering comparing the proposed scheme to the Leader algorithm. As can be seen from the DSSP annotations, the kinetic ordering of states is similar in both cases. The data for the Leader algorithm appear much more noisy because the DSSP strings for each mesostate are derived from the microstate that originally spawned the mesostate, and that is not necessarily a good representative of the actual centroid (see Fig. 2). This potential mismatch between properties of the first microstate *vs*. the added microstates does not mean, however, that drastically different microstates are combined into a single cluster. To show this, Fig. S3 plots the same data using the maximum likelihood estimate of the DSSP assignment string based on the underlying distribution of snapshots constituting a given mesostate. In Fig. S3, the DSSP maps between algorithms become very similar, and – as expected – resemble very much the original maps for the proposed algorithm as seen in Fig. 6.

The cFEPs themselves agree qualitatively well in that the basin of folded states encompasses about 38 % of the data. This is followed by a region with increasingly disordered states interspersed by a few enthalpic basins. The kinetically most distant states are helix-rich, and here quantitative agreement between algorithms is best. The most remarkable deviation is the depletion of the first barrier for the Leader algorithm. It shows that the proposed scheme is not only more efficient, but also provides a better mesostate partitioning. A higher barrier means that the number of transitions between mesostates on different sides of the barrier is lower, which most likely results from reduced mesostate overlap. In high-dimensional spaces (in contrast to the results for *n*-butane shown in Fig. 5), the Leader algorithm essentially introduces kinetic shortcuts by placing mesostates not in accordance with local density, but arbitrarily. This leads to the actual barrier crossing being obscured if structural distances to either side are comparatively small. The latter point is illustrated by the congruence between both algorithms in describing the barrier separating helix-rich states form the remainder. Here, structural differences as measured by RMSD are large and the same result is obtained for both algorithms. Panel B shows that the likelihood of observing such shortcuts can depend on the chosen measure of similarity. For dihedral

angles the density distribution in phase space is obviously different, and – in this particular case – results in larger quantitative differences between the two algorithms despite the qualitative nature of the cFEP and the kinetic ordering being preserved both with respect to each other and with respect to the RMSD-based network.

Parameter dependencies and robustness of derived mesostate networks for real-world example

One may ask whether the lack of congruence between algorithms in Fig. 6 (that was not observed in Fig. 5 presumably due to much lower dimensionality) now also implies a dependency on H. This is explored in Fig. S4, where it is shown that both changes in H can give rise to minor, unsystematic deviations that are, however, small in magnitude compared to the deviations seen between Leader algorithm and the proposed scheme. Along similar lines, the last question we explore is how robust results are upon changing the threshold size of mesostates, t_1. It is expected that for larger values of t_1 the density-based location of mesostate centroids will prove increasingly beneficial when comparing the proposed scheme to the Leader algorithm. In essence, the range of accessible conformations grows extremely quickly with t_1 in high-dimensional spaces, and blind placement may well create a mesostate that spans or extends into a barrier. Fig. 7 shows that this is precisely the case for the data on the β-sheet miniprotein (*DS5*) explored in Fig. 6. While at $t_1 = 0.27$ both algorithms generate cFEPs that share similar qualitative features and allow the identification of the same number of basins, the deterioration in information content is much less dramatic for the proposed algorithm (Panel A) compared to Leader (Panel B). For instance, at $t_1 = 0.32$, the data in Panel A resolve the same details as at finer resolution, whereas in Panel B all structure in the left half of the plot is missing. Interestingly, in this case the tightness of mesostates is no longer consistently higher for the proposed scheme and the total number of mesostates is no longer necessarily larger even though $H = 16$ throughout. In fact, the summary statistics reported in Table S1 can hardly explain the dramatic differences seen in Fig. 7. The similarity in overall statistics means that differences must be almost entirely on account of the anticipated superiority of the proposed scheme in situating mesostates appropriately in high-density regions when D is large. For instance, the number of microstates in the largest mesostate is up to an order of magnitude larger for the proposed scheme, and shows much more systematic changes with t_1 (Table S1). Lastly, similar to the case for Fig. 6, Fig. S5 presents the same data as Fig. 7 with the exception that the DSSP strings utilized to create the color traces in the upper parts of the plots are recomputed as maximum likelihood guesses over all members of each respective mesostate. Fig. S5 shows that the kinetic ordering is reasonably well-preserved even for the rather featureless cFEP for the Leader

algorithm at $t_1 = 0.40$, and more importantly that the centroid description is nearly indistinguishable from the maximum likelihood guess for the proposed scheme.

DISCUSSION AND CONCLUSION

In this contribution, we have presented a novel algorithm for the efficient construction of mesostate networks from (bio)molecular simulation data. The scheme adopts its architecture and some of its ideas from the BIRCH clustering algorithm[46] that is optimized for spatial constraints and low dimensionality.[56] One may ask whether the broad literature available on the subject contains alternative solutions to the problem as stated in the Introduction. The main issue in identifying appropriate algorithms is that few approaches state the problem exactly in identical fashion; for example, we require mesostates (clusters) to be homogeneous, non-overlapping, and of controllable size, whereas typical ways of posing the problem focus on allowing arbitrary cluster shapes and sizes.[57] Here, we will briefly discuss different classes of algorithms explicitly, and touch upon the reasons why they may violate one or more of our peculiar requirements. Readers are referred to the excellent review by Xu and Wunsch[56] for further details.

First, density-based algorithms such as DBSCAN,[58] DENCLUE,[59] or OPTICS[60] employ a local density threshold criterion to delineate regions of high density (clusters) from those of low density (background). These techniques will often fail to work in high-dimensional spaces due to inhomogeneous density distributions, and all violate our requirements for mesostates to be of homogeneous size and for all data to be important. Second, many established partitioning algorithms such as the aforementioned *K*-medoids or similar algorithms[61,62] scale unfavorably with dataset size if the desired number of mesostates is large. This remains true for many improved variants,[63,64] and consequently they are of little use when applied to very large simulation datasets irrespective of their individual virtues. Moreover, the stipulation to provide the number of clusters K upfront is inconvenient, as a *priori* it is not possible to relate K to a mesostate volume. Third, algorithms that explicitly impose an underlying class of distribution functions onto the data such as popular variants of the expectation-maximization scheme[65] require the data to conform approximately to the assumed shape. Fourth, projection-based approaches utilize information either in lower-dimensional subspaces (such as in CLIQUE,[66] OptiGrid,[67] or the very recent Halite algorithm[68]), or try to improve cluster separability by increasing data dimensionality coupled to the so-called kernel trick (for instance in support vector clustering[69]). These are both promising strategies for analyzing molecular simulation

data, but involve a fair number of parameters, and are not always easy to use by nonspecialist researchers. In addition, dimensionality reduction techniques bear the danger of introducing kinetic shortcuts into derived networks, whereas dimensionality increases may be difficult to keep computationally tractable. Fifth, amongst grid-based approaches there are those that do not scale up to large values of D such as WaveCluster.[70] In addition, there are several grid-based methods including CLIQUE[66] and OptiGrid[67] that manage to overcome the usual inapplicability of grid-based approaches to cases when D is large. However, CLIQUE scales poorly with data dimensionality ($\sim D^2$) in time, whereas OptiGrid leaves choices for required heuristics open. Overall, the sheer number of proposed approaches and the large overlap between them highlight again the fact that the problem of clustering can be posed any number of ways. Moreover, nearly all algorithms are reported to outperform earlier counterparts, which makes a quick evaluation of their potential weaknesses difficult.

In conclusion, the proposed algorithm (Fig. 1) was specifically designed to deal with the challenges posed by coarse-graining molecular simulation data into networks of mesostates that preserve important information regarding free energy basins and barriers. It has the following properties:

I. It operates in near-linear time with respect to dataset size (Fig. 4).

II. Its results are not strongly dependent on the choice of input parameters, *i.e.*, primarily the tree height H (Figs. 3, 5, and S4). The choice for t_H and the interpolation scheme are coupled to the chosen H, but thus far linear interpolation and choosing t_H to match approximately the maximum distance in the data have proven sufficient.

III. It creates mesostates that are of consistent size (set by t_1 and H) and free of overlap (Figs. 2, 3, and S1).

IV. Mesostates track local density well, which is essential for describing network connectivity (kinetics) in high-dimensional spaces in authentic fashion (Figs. 6-7, S3, and S5).

We believe that the algorithm will be useful to the biomolecular simulation community. It has been implemented in the open source software project CAMPARI,[71] and a current development snapshot of the source files is available upon request via campari.software@gmail.com. We did not specifically look for other problem domains to apply the proposed scheme to, but it may well prove suitable to applications with similar criteria. Lastly, ongoing research is concerned with utilizing the inherent multi-resolution nature of the output of the algorithm to incorporate and extract kinetic information directly, and with the development of inexpensive measures of network robustness and quality.

ASSOCIATED CONTENT

Supporting Information. Derivation of efficient formulas for computing intrinsic and relative cluster properties. Implementation of other clustering algorithms. Moments of inertia as fluctuating weights. Table (S1) detailing statistics regarding data underlying Fig. 7. Supporting figures (S1-S5) on mesostate boundaries (Fig. S1), *mfpt*-correlation analysis for *n*-butane (Fig. S2), robustness of results on Beta3S (*DS5*) as a function of *H* (Fig. S4), and alternate versions of Figs. 6 and 7 (Figs. S3 and S5). This material is available free of charge via the Internet at http://pubs.acs.org.

ACKNOWLEDGMENT

This work was partially supported by a grant from the Swiss National Science Foundation to A.C. and a personal grant from the Forschungskredit of the University of Zurich to A.V.

REFERENCES

1. Jain, A. K.; Murty, M. N.; Flynn, P. J. *ACM Comput. Surv.* **1999**, *31*, 264–323.
2. Pan, A. C.; Roux, B. *J. Chem. Phys.* **2008**, *129*, 64107.
3. Yao, Y.; Sun, J.; Huang, X.; Bowman, G. R.; Singh, G.; Lesnick, M.; Guibas, L. J.; Pande, V. S.; Carlsson, G. *J. Chem. Phys.* **2009**, *130*, 144115.
4. Noé, F.; Schütte, C.; Vanden-Eijnden, E.; Reich, L.; Weikl, T. R. *Proc. Natl. Acad. Sci. USA.* **2009**, *106*, 19011–19016.
5. Schön, J. C.; Sibani, P. *Europhys. Lett.* **2000**, *49*, 196–202.
6. Smith, L. J.; Daura, X.; van Gunsteren, W. F. *Prot. Struct. Func. Bioinf.* **2002**, *48*, 487–496.
7. Grossfield, A.; Feller, S. E.; Pitman, M. C. *Prot. Struct. Func. Bioinf.* **2007**, *67*, 31–40.
8. Bowman, G. R.; Ensign, D. L.; Pande, V. S. *J. Chem. Theory Comput.* **2010**, *6*, 787–794.
9. Scalco, R.; Caflisch, A. *J. Phys. Chem. B.* **2011**, *115*, 6358–6365.
10. Li, A.; Daggett, V. *J. Mol. Biol.* **1998**, *275*, 677–694.
11. Rao, F.; Caflisch, A. *J. Mol. Biol.* **2004**, *342*, 299–306.
12. Buchete, N. V.; Hummer, G. *J. Phys. Chem. B.* **2008**, *112*, 6057–6069.
13. Gfeller, D.; Los Rios, P. de; Caflisch, A.; Rao, F. *Proc. Natl. Acad. Sci. USA.* **2007**, *104*, 1817–1822.
14. Swope, W. C.; Pitera, J. W.; Suits, F. *J. Phys. Chem. B.* **2004**, *108*, 6571–6581.

15. Groot, B. de; Daura, X.; Mark, A. E.; Grubmüller, H. *J. Mol. Biol.* **2001**, *309*, 299–313.
16. Schütte, C.; Fischer, A.; Huisinga, W.; Deuflhard, P. *J. Comp. Phys.* **1999**, *151*, 146–168.
17. Krivov, S. V.; Muff, S.; Caflisch, A.; Karplus, M. *J. Phys. Chem. B.* **2008**, *112*, 8701–8714.
18. Ramanathan, A.; Yoo, J. O.; Langmead, C. J. *J. Chem. Theory Comput.* **2011**, *7*, 778–789.
19. Chodera, J. D.; Singhal, N.; Pande, V. S.; Dill, K. A.; Swope, W. C. *J. Chem. Phys.* **2007**, *126*, 155101.
20. Keller, B.; Daura, X.; van Gunsteren, W. F. *J. Chem. Phys.* **2010**, *132*, 74110.
21. Shao, J.; Tanner, S. W.; Thompson, N.; Cheatham III, T. E. *J. Chem. Theory Comput.* **2007**, *3*, 2312–2334.
22. Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. *Annu. Rev. Phys. Chem.* **2002**, *53*, 291–318.
23. Wales, D. J. *Mol. Phys.* **2002**, *100*, 3285–3305.
24. Carr, J. M.; Wales, D. J. *J. Phys. Chem. B.* **2008**, *112*, 8760–8769.
25. Carr, J. M.; Wales, D. J. *Phys. Chem. Chem. Phys.* **2009**, *11*, 3341–3354.
26. Stillinger, F. H.; Weber, T. A. *Phys. Rev. A.* **1982**, *25*, 978–989.
27. Carr, J. M.; Trygubenko, S. A.; Wales, D. J. *J. Chem. Phys.* **2005**, *122*, 234903.
28. Muff, S.; Caflisch, A. *Prot. Struct. Func. Bioinf.* **2008**, *70*, 1185–1195.
29. Bellman, R. E. Dynamic programming; Dover: Mineola, NY, USA, 2003; p ix
30. Chávez, E.; Navarro, G. *Inf. Process. Lett.* **2003**, *85*, 39–46.
31. Radovanović, M.; Nanopoulos, A.; Ivanović, M. *J. Mach. Learn. Res.* **2010**, *11*, 2487–2531.
32. Jain, A.; Zongker, D. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 153–158.
33. Pearson, K. *Philos. Mag. 6th Ser.* **1901**, *2*, 559–572.
34. Comon, P. *Signal Process.* **1994**, *36*, 287–314.
35. Roweis, S. T.; Saul, L. K. *Science.* **2000**, *290*, 2323–2326.
36. Hartigan, J. A. Clustering algorithms; Wiley: New York, NY, USA, 1975; pp 74-78
37. Prinz, J.-H.; Wu, H.; Sarich, M.; Keller, B.; Senne, M.; Held, M.; Chodera, J. D.; Schütte, C.; Noé, F. *J. Chem. Phys.* **2011**, *134*, 174105.
38. Muff, S.; Caflisch, A. *J. Chem. Phys.* **2009**, *130*, 125104.
39. Convertino, M.; Vitalis, A.; Caflisch, A. *J. Biol. Chem.* **2011**, *286*, 41578–41588.
40. Gonzalez, T. F. *Theor. Comput. Sci.* **1985**, *38*, 293–306.
41. Dasgupta, S.; Long, P. M. *J. Comp. Sys. Sci.* **2005**, *70*, 555–569.

42. Bowman, G. R.; Beauchamp, K. A.; Boxer, G.; Pande, V. S. *J. Chem. Phys.* **2009**, *131*, 124101.
43. Day, W. H. E. *J. Classif.* **1984**, *1*, 7–24.
44. Murtagh, F. *Comput. J.* **1983**, *26*, 354–359.
45. Torda, A. E.; van Gunsteren, W. F. *J. Comput. Chem.* **1994**, *15*, 1331–1340.
46. Zhang, T.; Ramakrishnan, R.; Livny, M. In *SIGMOD `96: Proceedings of the 1996 ACM SIGMOD international conference on management of data,* Montreal, QC, Canada, June 04 - June 06, 1996; Widom, J., ed; ACM Press: New York, NY, USA, 1996; pp 103-114.
47. Ganti, V.; Ramakrishnan, R.; Gehrke, J.; Powell, A.; French, J. In *Proceedings of the 15th International Conference on Data Engineering,* Sydney, NSW, Australia, March 23 - March 26, 1999; Kitsuregawa, M., Maciaszek, L., Papazoglou, M., Pu, C., eds; IEEE Computer Society: Los Alamitos, CA, USA, 1999; pp 502-511.
48. Karpen, M. E.; Tobias, D. J.; Brooks III, C. L. *Biochemistry.* **1993**, *32*, 412–420.
49. Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. *J. Am. Chem. Soc.* **1996**, *118*, 11225–11236.
50. Jensen, C. H.; Nerukh, D.; Glen, R. C. *J. Chem. Phys.* **2008**, *128*, 115107.
51. Guarnera, E.; Pellarin, R.; Caflisch, A. *Biophys. J.* **2009**, *97*, 1737–1746.
52. Krivov, S. V.; Karplus, M. *Proc. Natl. Acad. Sci. USA.* **2008**, *105*, 13841–13846.
53. Krivov, S. V.; Karplus, M. *J. Phys. Chem. B.* **2006**, *110*, 12689–12698.
54. Krivov, S. V.; Karplus, M. *J. Chem. Phys.* **2002**, *117*, 10894–10903.
55. Evans, D. A.; Wales, D. J. *J. Chem. Phys.* **2003**, *118*, 3891–3897.
56. Xu, R.; Wunsch II, D. *IEEE Trans. Neural Netw.* **2005**, *16*, 645–678.
57. Zhang, J.; Hsu, W.; Lee, M. L. *J. Intell. Inf. Syst.* **2005**, *24*, 5–27.
58. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining,* Portland, Oregon, USA, August 2 - August 4, 1996; Simoudis, E., Han, J., Fayyad, U., eds; AAAI Press: Menlo Park, CA, USA, 1996; pp 226-231.
59. Hinneburg, A.; Keim, D. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining,* New York City, NY, USA, August 27 - August 31, 1998; Agrawal, R., Stolorz, P., eds; AAAI Press: Menlo Park, CA, USA, 1998; pp 58-65.
60. Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; Sander, J. In *SIGMOD `99: Proceedings of the 1999 ACM SIGMOD international conference on management of data,* Philadelphia, PA, USA, May 31 - June 03, 1999; Clifford, J., King, R., eds; ACM Press: New York, NY, USA, 1999; pp 49-60.
61. Carpenter, G. A.; Grossberg, S.; Rosen, D. B. *Neural Networks.* **1991**, *4*, 493–504.

62. Moore, B. In *Proceedings of the 1988 Connectionist Models Summer School,* Carnegie Mellon University, June 17 - June 26, 1988; Touretzky, D. S., Hinton, G. E., Sejnowski, T. J., eds; Morgan Kaufmann: San Mateo, CA, USA, 1989; pp 174-185.
63. Li, M. J.; Ng, M. K.; Cheung, Y.-M.; Huang, J. Z. *IEEE Trans. Knowl. Data Eng.* **2008**, *20*, 1519–1534.
64. Ng, R. T.; Han, J. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 1003–1016.
65. Dempster, A. P.; Laird, N. M.; Rubin, D. B. *J. Roy. Stat. Soc. B Meth.* **1977**, *39*, 1–38.
66. Agrawal, R.; Gehrke, J.; Gunopulos, D.; Raghavan, P. *Data Min. Knowl. Discov.* **2005**, *11*, 5–33.
67. Hinneburg, A.; Keim, D. A. In *Proceedings of the 25th VLDB Conference,* Edinburgh, Scotland, September 7 - September 10, 1999; Atkinson, M. P., Orlowska, M. E., Valduriez, P., Zdonik, S. B., Brodie, M. L., eds; Morgan Kaufmann: San Francisco, CA, USA, 1999; pp 506-517.
68. Cordeiro, R. L. F.; Traina, A. J. M.; Faloutsos, C.; Traina Jr., C. *IEEE Trans. Knowl. Data Eng.* **2011**, DOI: 10.1109/TKDE.2011.176.
69. Ben-Hur, A.; Horn, D.; Siegelmann, H. T.; Vapnik, V. *J. Mach. Learn. Res.* **2001**, *2*, 125–137.
70. Sheikholeslami, G.; Chatterjee, S.; Zhang, A. In *Proceedings of the 24th VLDB Conference,* New York City, New York, USA, August 24 - August 27, 1998; Gupta, A., Shmueli, O., Widom, J., eds; Morgan Kaufmann: San Francisco, CA, USA, 1999; pp 428-439.
71. Vitalis, A.; Steffen, A.; Lyle, N.; Mao, A. H.; Pappu, R. V. (2010) CAMPARI v1.0; http://sourceforge.net/projects/campari (accessed December 21, 2011)
72. Kabsch, W.; Sander, C. *Biopolymers.* **1983**, *22*, 2577–2637.

FIGURE CAPTIONS

Figure 1: **Schematic illustration of the proposed algorithm on an arbitrary 2D dataset.** The example shown uses $H = 3$. At each level indicated by the gray rectangles, the dataset is shown as a scatter plot where colors indicate mesostate assignment (colors repeat at the bottom level). At the root, all data are lumped together and their centroid is indicated by the black octahedron. At the three actual levels, the centroids of the 5, 14, and 66 mesostates, respectively, are highlighted by colored octahedra. Parent-child relationships are indicated by lines colored according to the parent. Two issues at the level corresponding to t_2 are indicated that both stem from the fact that for every level except the leaf level, essentially only one single scan of the data is used. First, it is possible for two mesostates centroids to be extremely close to one another (overlap) on account of a split in pathways further toward the root. Second, some mesostates may end up without any children, even though at least one microstate was nearby. Such problems are largely eliminated at the leaf level, which is why the two-pass strategy is essential. In general, mesostate centroids track local data density well, and no major partitioning errors are seen at any of the levels.

Figure 2: **Clustering results for the 2-dimensional dataset *DS1*.** Data correspond to 25000 snapshots of two principal components obtained from a high-dimensional dataset. **A**: Scatter plot of entire dataset along with contour lines delineating regions of high density (based on a 2D data histogram with bin-widths of 1.77 and 1.07 Å, respectively). **B**: Using the proposed algorithm with $H = 4$, $t_1 = 5$ Å, $t_4 = 25$ Å, we obtained 280 mesostates at level 1 (6 of them containing only a single microstate). The members of the largest 30 of them are shown as dots in different colors. Since some colors are similar, filled circles denoting the centroid position of each mesostate are added. Density contours are overlaid. Note the different axis scaling compared to A. **C**: The same as B for the simple Leader algorithm (see Supporting Methods). Here we obtained 195 mesostates (a single one containing only a single microstate). Filled circles denote the microstate serving as the cluster center and not the actual centroid. **D**: The same as B for the rigorous agglomerative scheme with mean linkage (see Supporting Methods). This algorithm yielded 218 mesostates with 10 of them consisting of a single microstate only. The mesostate size thresholds were applied consistently for all three cases. The top 30 clusters contained 44.7, 58.3, and 57.7% of *DS1* for Panels B, C, and D, respectively.

Figure 3: **Dependence of clustering results on tree height *H* for *DS2* (750000 snapshots).** The proposed algorithm was used with variable H, $t_1 = 1.5$ Å, and $t_H = 8$ Å. **A**: The total number of proposed mesostates is plotted along with the number of mesostates consisting of only a single microstate. Also,

the total number of microstates contained in mesostates of size 10 or larger is shown along with the total number of such mesostates. **B**: CPU time does not show a similar dependency on H to any of the other quantities. The reported times do no contain contributions from actual dataset I/O, but do contain contributions from computing cluster properties, and for writing graph and network files. Note the logarithmic scale chosen to aid clarity (right y-axis). The black dashed line is obtained as a fit of the original t_{CPU}-values to H using data from $H \geq 12$. The increasing tightness of clusters is expected based on Panel A and shown as well (linear scale). Tightness is measured as the average, normalized distance of each microstate to the centroid of its corresponding mesostate. Only those mesostates are included that contain more than one microstate.

Figure 4: **Scaling of computational cost with dataset size for *DS3*.** The threshold value (t_1) was 0.3 in either case. For the proposed algorithm, we used as additional settings $H = 16$ and $t_{16} = 1.0$. **A**: Elapsed CPU times for clustering are shown as a function of dataset size, N. See caption to Fig. 3 for details on CPU times. Results for the Leader algorithm are distinctively nonlinear, whereas a very good line fit can be obtained to describe the results for the proposed scheme. **B**: The same data in a double logarithmic plot. Congruent with the results in Panel A, line fits to both sets of data reveal scaling exponents of 1.68 and 1.06 for Leader and proposed algorithm, respectively.

Figure 5: **Cut-based free energy profiles for *n*-butane.** Data are clustered with several algorithms and based on either dihedral angle distances (Panel A), or on all-atom RMSD values of Cartesian coordinates (Panel B). Threshold settings used were $t_1 = 7°$ and $t_1 = 0.12$ Å, respectively, with $t_H = 100°$ and $t_H = 1.0$ Å as coarsest criteria for the proposed algorithm. The lower part of each panel shows the actual cFEPs for six different methods. The labels "Hierar.", and "L-Fwd" denote the rigorous agglomerative scheme, and the Leader algorithm with both search directions flipped, respectively (see Supporting Methods). Results for the proposed scheme are shown for three different values of H. Green dashed lines indicate the positions of prominent barriers in the cFEPs (see text), and are placed identically in both panels. The top half of each plot shows traces for three different algorithms that each depict the coarse state assignment for the three dihedral angles in the system in correspondence with the progress variable of the cFEP. The first (Leader), or otherwise the central microstate of each mesostate was used to derive the state assignment. Colors extend along the abscissa in accordance with mesostate weights. The term E in the cut-based free energy corresponds to the total number of microstate transitions, *i.e.*, 49999. Only the 2000 largest mesostates are actually plotted in each case to keep the number of objects displayed tractable. This does not noticeably alter the appearance of the

figure at typical resolution/enlargement.

Figure 6: **Cut-based free energy profiles for beta3S (*DS5*).** Data are clustered with two algorithms and based on either RMSD values of backbone nitrogen and oxygen atoms over residues 3-18 ($D = 96$, Panel A), or based on ϕ,ψ,ω-angles over residues 3-18 ($D = 48$, Panel B). Threshold settings used were $t_1 = 1.5$ Å and $t_1 = 25°$, respectively, with $t_H = 10.0$ Å and $t_H = 100°$ as coarsest criteria for the proposed algorithm. The bottom half of each panel shows cFEPs similar to Fig. 5. Dashed lines correspond to positions of dominant barriers identified in Panel A. The top half of each plot shows traces for both algorithms that each depict the DSSP letter assignment[72] for the 20 residues in the system in correspondence with the progress variable of the cFEP. The first (Leader), or the microstate nearest to the centroid of each mesostate was used to derive the DSSP string. The significantly larger amount of noise in these maps for the Leader-derived data stems from mesostate overlap and the poorly defined relationship between mesostate centroid and the microstate that spawned it. Colors extend along the abscissa in accordance with mesostate weights. In addition to the colors identified in the legend, unassigned residues (white in the plots) are interpreted to correspond to extended coil states. E was 999999, and only the 7500 largest mesostates are actually plotted (see Fig. 5 regarding pruning).

Figure 7: **Impact of t_1 on cut-based free energy profiles for beta3S (*DS5*).** To facilitate fast and correct computation of cluster properties, the sine/cosine values of the ϕ,ψ,ω-angles over residues 3-18 ($D = 96$) served as input data (compare Panel B in Fig. 6). Data are clustered either with the proposed scheme using $H = 16$ and $t_H = 1.0$ (Panel A), or with the Leader algorithm (Panel B). Nine different values for t_1 ranging from 0.27 to 0.40 were explored (see Table S1 for associated network statistics). DSSP traces are shown in analogy to Fig. 6 for the case of $t_1 = 0.40$ to highlight the differences in robustness between algorithms. Results for $t_1 = 0.27$ for both algorithms are plotted in both panels to facilitate direct comparisons. E was 999999, and only the 7500 largest mesostates in each case are actually plotted (see Fig. 5 regarding pruning).

Table 1. Normalized accuracies of simplified computations of cluster properties for *DS2*.

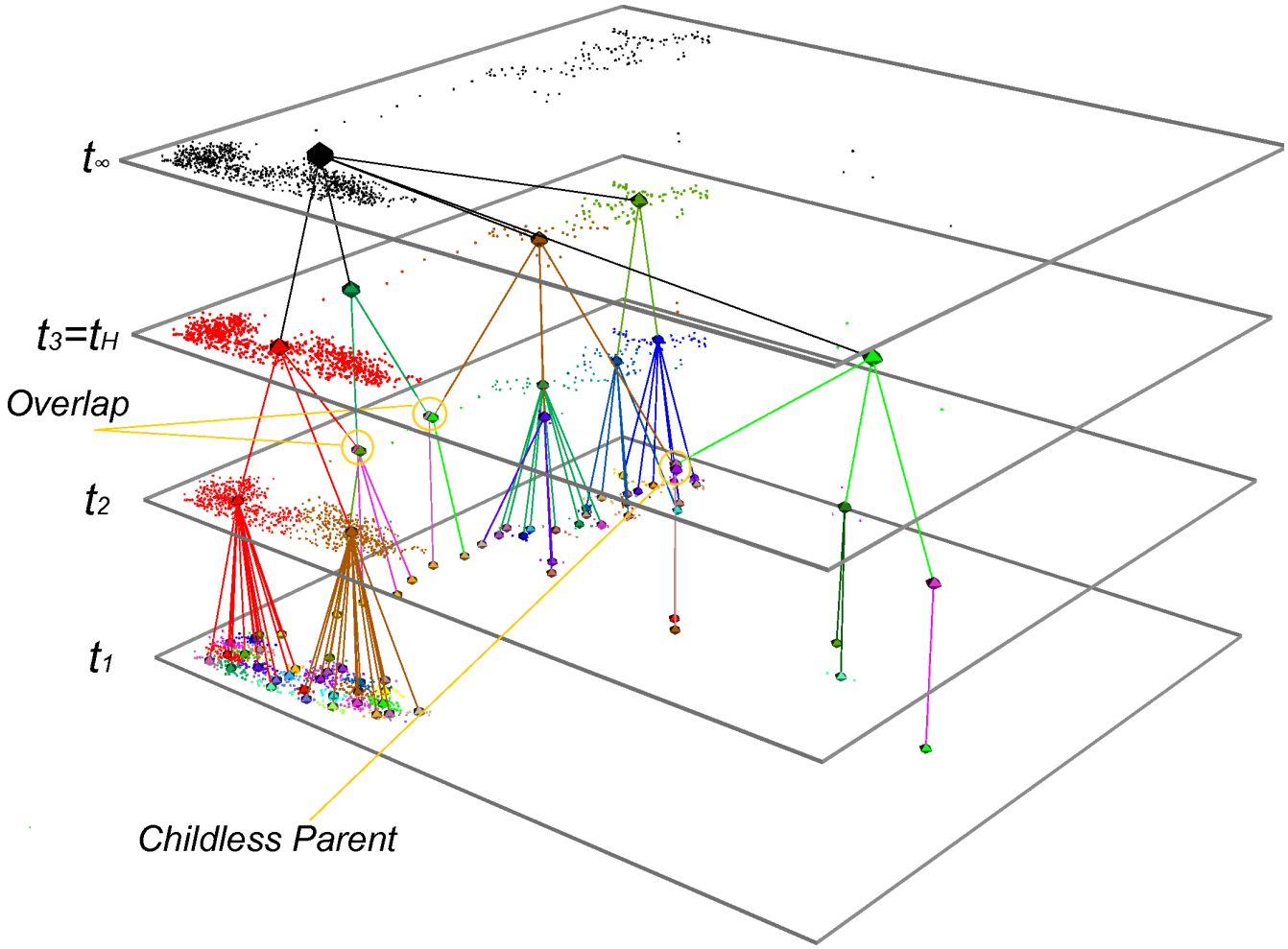
Measure	***D***	t_1	$N_{c\geq 3}$	$L_2(d_c)/t_1$	$L_\infty(d_c)/t_1$	$L_2(r_c)/t_1$	$L_\infty(r_c)/t_1$	$L_2(d_{IC})/t_1$	$L_\infty(d_{IC})/t_1$
RMSD	66	1.6Å	3554	$1.4\text{x}10^{-3}$	*$2.2\text{x}10^{-2}$*	$2.3\text{x}10^{-4}$	$3.1\text{x}10^{-3}$	$1.0\text{x}10^{-2}$	***$1.6\text{x}10^{-1}$***
ω,ϕ,ψ	33	25.0º	3123	$2.6\text{x}10^{-5}$	$1.4\text{x}10^{-3}$	$1.6\text{x}10^{-14}$	$1.9\text{x}10^{-13}$	*$4.9\text{x}10^{-2}$*	***$2.5\text{x}10^{-1}$***
sincos	66	0.27	2641	$1.2\text{x}10^{-15}$	$1.1\text{x}10^{-14}$	$6.6\text{x}10^{-16}$	$3.0\text{x}10^{-15}$	$6.7\text{x}10^{-16}$	$4.1\text{x}10^{-15}$
D-RMS	144	1.7Å	3648	$1.6\text{x}10^{-14}$	$2.0\text{x}10^{-13}$	$1.0\text{x}10^{-14}$	$1.0\text{x}10^{-13}$	$9.2\text{x}10^{-15}$	$6.8\text{x}10^{-14}$
ω,ϕ,ψ / I	33	22.0º	3365	*$3.5\text{x}10^{-2}$*	*$8.7\text{x}10^{-2}$*	$3.1\text{x}10^{-3}$	*$3.5\text{x}10^{-2}$*	*$4.9\text{x}10^{-2}$*	***$2.6\text{x}10^{-1}$***
sincos / I	66	0.24	2931	$4.3\text{x}10^{-3}$	*$2.7\text{x}10^{-2}$*	$2.7\text{x}10^{-3}$	*$1.5\text{x}10^{-2}$*	$7.2\text{x}10^{-3}$	*$7.2\text{x}10^{-2}$*

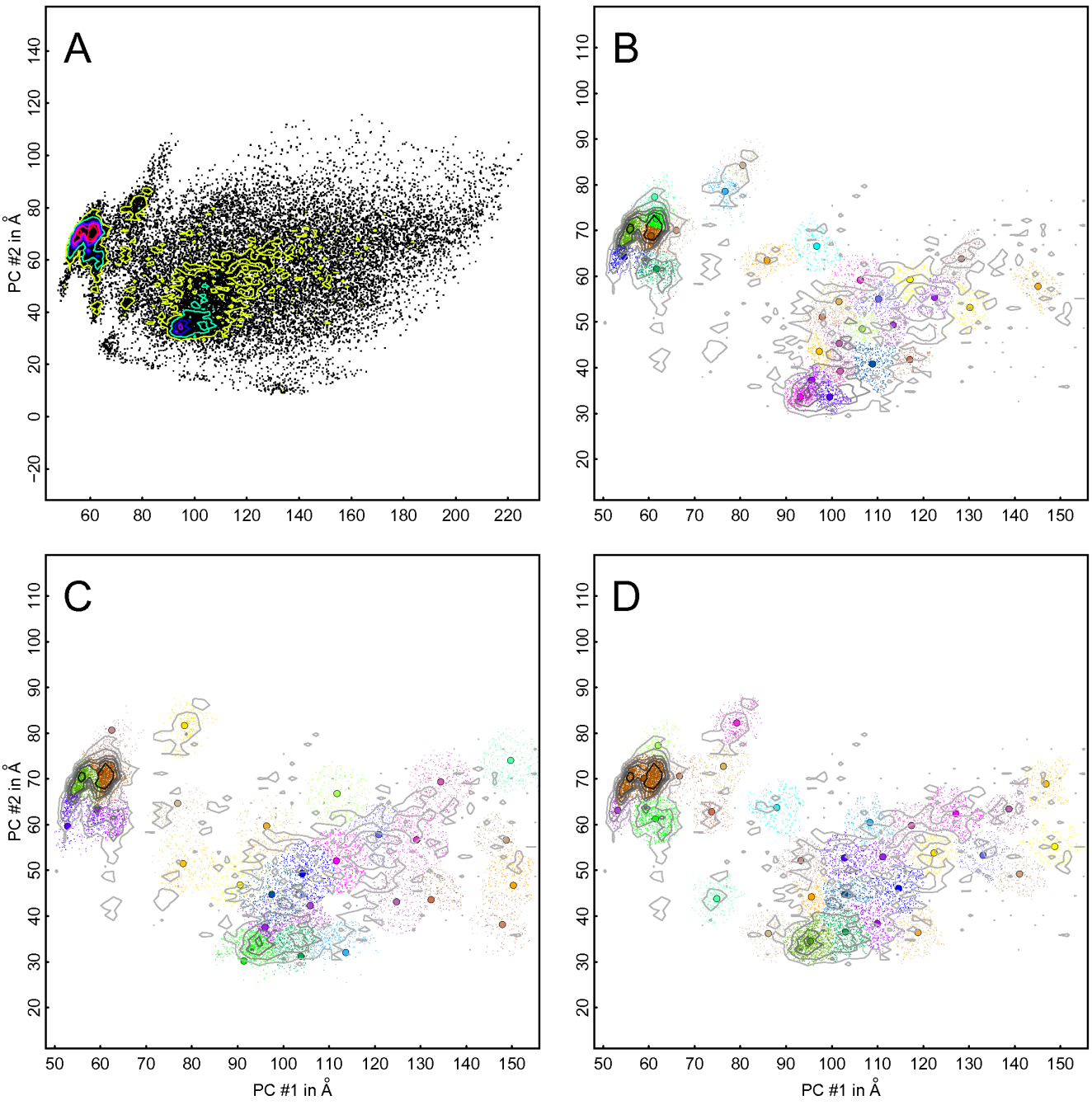
RMSD utilized Cartesian positions of 22 atoms and quaternion-based alignment. The D-RMS is a set of partially redundant interatomic distances. "ω,ϕ,ψ" utilizes the 3 backbone dihedral angles of 11 consecutive residues, and "sincos" denotes the same data in sine/cosine space. The "/ I" denotes that each underlying torsional degree of freedom was subjected to a fluctuating weight corresponding to the moment of inertia associated with that torsion (see Supporting Methods). $N_{c\geq 3}$ stands for the number of identified clusters of size 3 or larger. The L_2 symbol stands for the quadratic norm of the difference between a cluster property computed using either one of equations 1, 2, S2, S5, S7, or exactly by enumerating it for all snapshots (RMS deviation). The L_∞ symbol corresponds to the associated L_∞ norm, *i.e.*, the largest deviation in the set. d_c and r_c were evaluated for all $N_{c\geq 3}$ clusters, while $d_{IC}(C,j)$ was computed for all $N_{c\geq 3}$ with respect to both a random snapshot and a snapshot from the same cluster. Italic font highlights differences between 1 and 10%, while bold italic font is used for those deviations exceeding 10%.

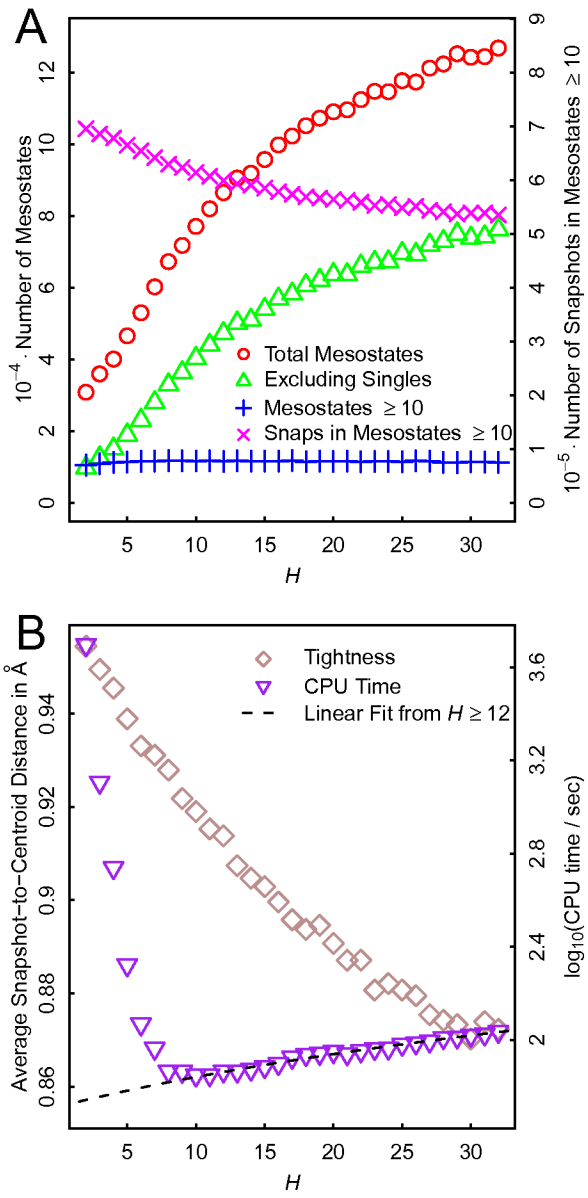
Table 2. Statistics for the data in Panel A of Fig. 5 that are based on *DS4* and dihedral angle distances.

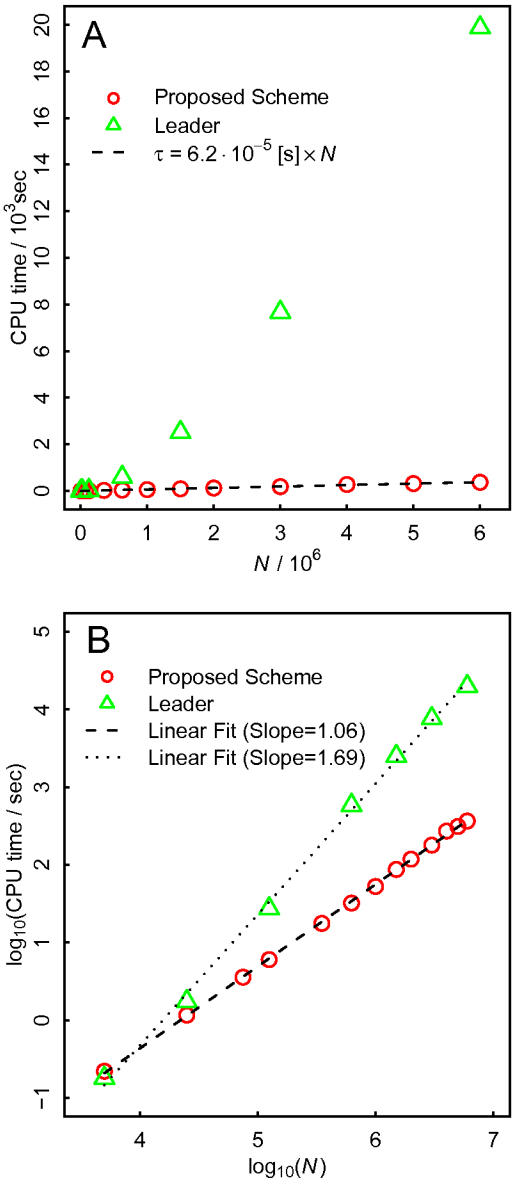
Algorithm	# Clusters	Mean(r_c)	F_V in %
Hierarchical	4338	6.37	23.8
Leader	5459	6.72	35.2
L-Fwd	5420	6.43	30.7
H=4	6348	5.71	25.2
H=8	7705	5.38	25.5
H=24	10161	4.92	25.7

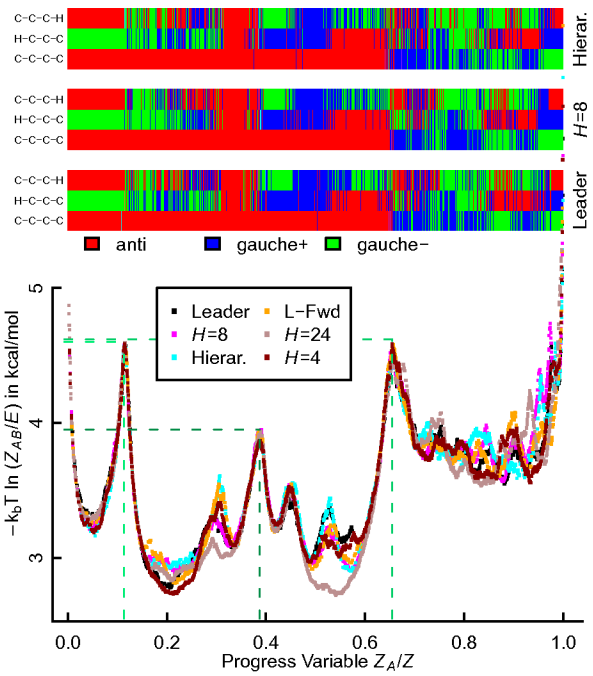
The total number of clusters (including those of size 1) for each algorithm is given in column 2. The Euclidean snapshot-centroid distance (r_c) averaged over all mesostates with at least two microstates is provided in degrees. From the mean radius, the fractional volume occupation is computed by assuming uniform density and spherical clusters as $F_V = N_c (4/3)\pi \left[(4/3) \langle r_c \rangle \right]^3 V_{total}^{-1}$. V_{total} is simply the phase space volume of 360º cubed ($D = 3$). Note that mesostates with only a single microstate are included in N_c, but do not contribute to $\langle r_c \rangle$.

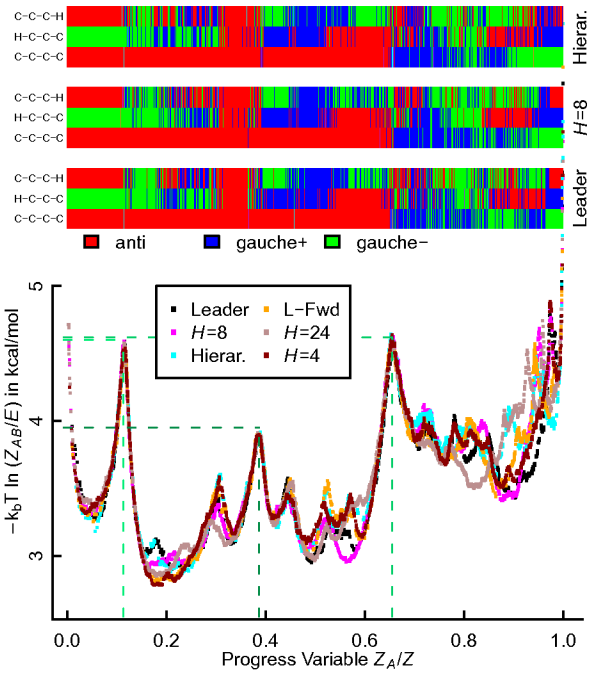

t_∞
$t_3=t_H$
Overlap
t_2
t_1
Childless Parent

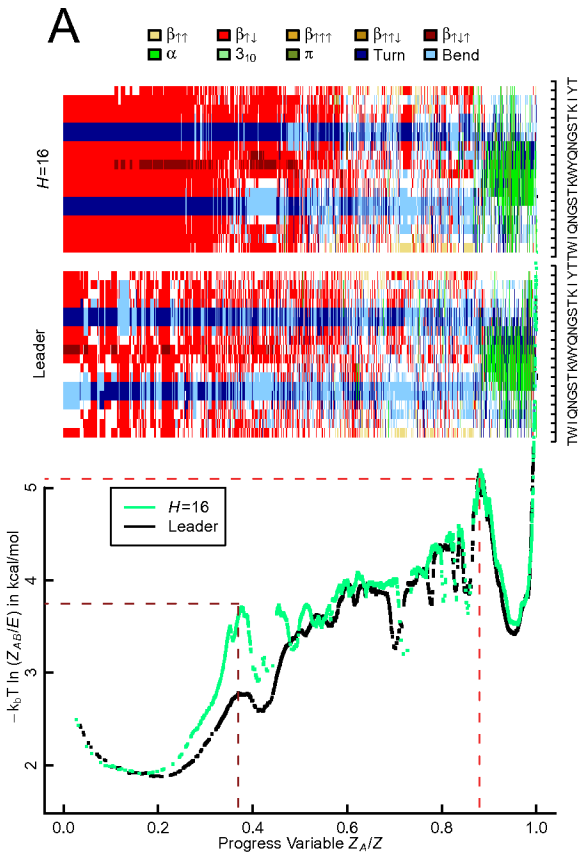

A
B
C
D
PC #2 in Å
PC #1 in Å

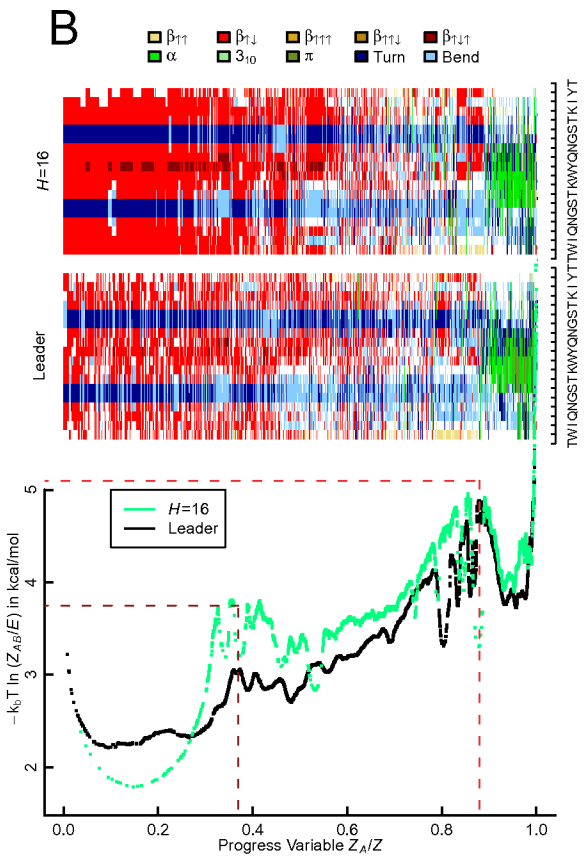

A
10^{-4} · Number of Mesostates
10^{-5} · Number of Snapshots in Mesostates ≥ 10
Total Mesostates
Excluding Singles
Mesostates ≥ 10
Snaps in Mesostates ≥ 10
H
B
Average Snapshot−to−Centroid Distance in Å
log10(CPU time / sec)
Tightness
CPU Time
Linear Fit from H ≥ 12
H

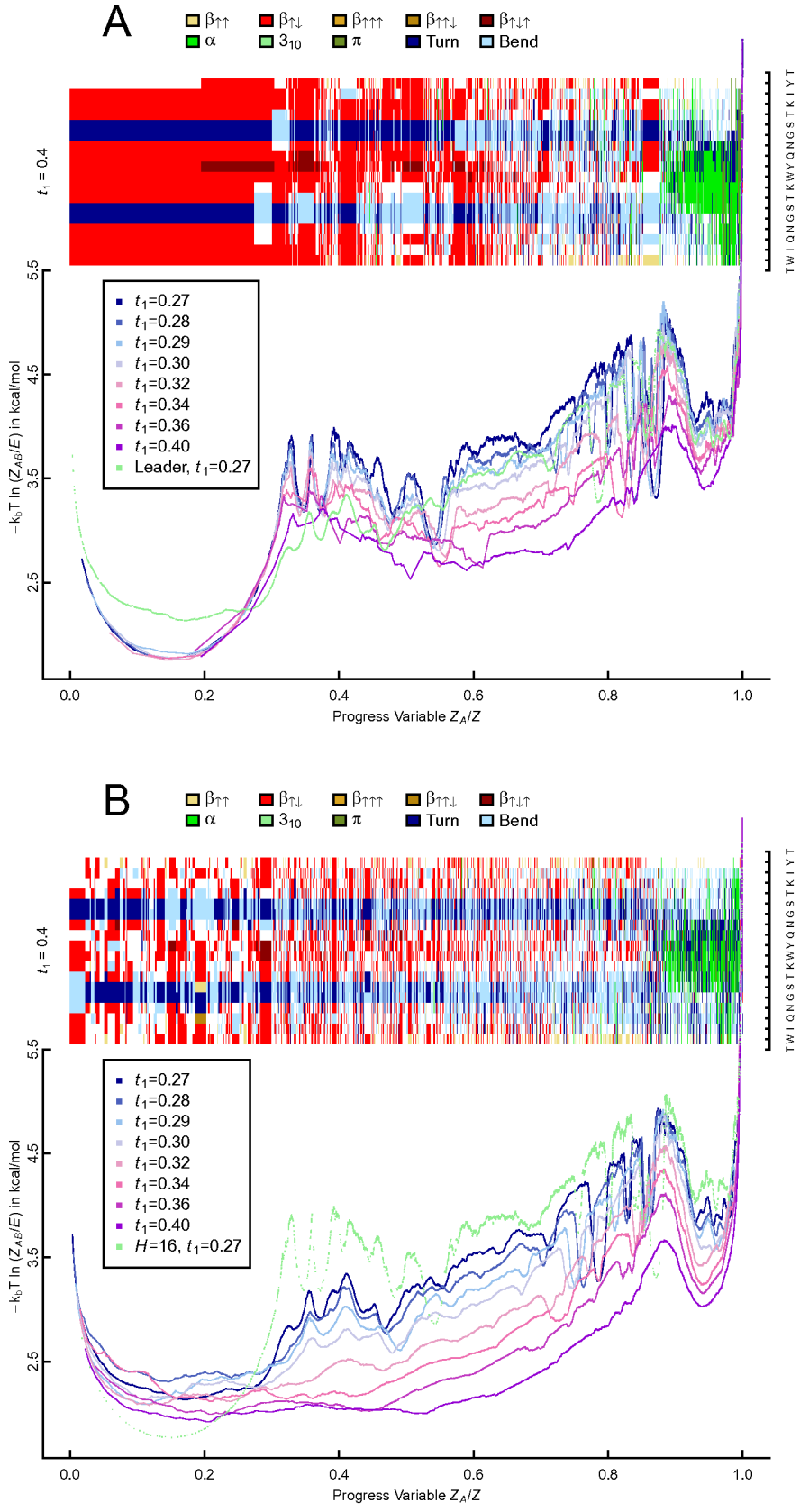

A
Proposed Scheme
Leader
$\tau = 6.2 \cdot 10^{-5}$ [s] × N
CPU time / 10^3sec
N / 10^6
B
Proposed Scheme
Leader
Linear Fit (Slope=1.06)
Linear Fit (Slope=1.69)
$\log_{10}$(CPU time / sec)
$\log_{10}(N)$


C–C–C–H
H–C–C–C
C–C–C–C
Hierar.
H=8
Leader
anti
gauche+
gauche−
Leader
H=8
Hierar.
L–Fwd
H=24
H=4
−k_bT ln (Z_AB/E) in kcal/mol
Progress Variable Z_A/Z


C–C–C–H
H–C–C–C
C–C–C–C
Hierar.
H=8
Leader
anti
gauche+
gauche−
Leader
H=8
Hierar.
L–Fwd
H=24
H=4
−k_bT ln (Z_AB/E) in kcal/mol
Progress Variable Z_A/Z


A
β↑↑
β↑↓
β↑↑↑
β↑↑↓
β↑↓↑
α
3₁₀
π
Turn
Bend
H=16
Leader
TWIQNGSTKWYQNGSTKIYTTWIQNGSTKWYQNGSTKIYT
H=16
Leader
−k₀T ln (Z_AB/E) in kcal/mol
Progress Variable Z_A/Z


B
β↑↑
β↑↓
β↑↑↑
β↑↑↓
β↑↓↑
α
3₁₀
π
Turn
Bend
H=16
Leader
TWIQNGSTKWYQNGSTKIYTTWIQNGSTKWYQNGSTKIYT
H=16
Leader
−k₀T ln (Z_AB/E) in kcal/mol
Progress Variable Z_A/Z


A
β↑↑
β↑↓
β↑↑↑
β↑↑↓
β↑↓↑
α
3₁₀
π
Turn
Bend
t₁ = 0.4
T W I Q N G S T K W Y Q N G S T K I Y T
t₁=0.27
t₁=0.28
t₁=0.29
t₁=0.30
t₁=0.32
t₁=0.34
t₁=0.36
t₁=0.40
Leader, t₁=0.27
−k₀T ln (Z_AB/E) in kcal/mol
2.5
3.5
4.5
5.5
0.0
0.2
0.4
0.6
0.8
1.0
Progress Variable Z_A/Z
B
β↑↑
β↑↓
β↑↑↑
β↑↑↓
β↑↓↑
α
3₁₀
π
Turn
Bend
t₁ = 0.4
T W I Q N G S T K W Y Q N G S T K I Y T
t₁=0.27
t₁=0.28
t₁=0.29
t₁=0.30
t₁=0.32
t₁=0.34
t₁=0.36
t₁=0.40
H=16, t₁=0.27
−k₀T ln (Z_AB/E) in kcal/mol
2.5
3.5
4.5
5.5
0.0
0.2
0.4
0.6
0.8
1.0
Progress Variable Z_A/Z